Cyclistic Trip Data

Thanh Nguyen

2022-12-28

R Markdown

This is my code notebook for the Cyclistic Trip Data Analysis, for Googld Data Analytics Capstone Projects So firstly I install and load packages

Import data files

I went to where I store my downloaded datasets, copied the paths and pasted them into R for importing. I used Macbook so it can be different for Windows users.

- jan2021 <- read_csv("/Users/…/202101-divvy-tripdata.csv")
- feb2021 <- read_csv("/Users/…/202102-divvy-tripdata.csv")
- mar2021 <- read_csv("/Users/…/202103-divvy-tripdata.csv")
- apr2021 <- read_csv("/Users/…/202104-divvy-tripdata.csv")
- may2021 <- read_csv("/Users/…/202105-divvy-tripdata.csv")
- jun2021 <- read_csv("/Users/…/202106-divvy-tripdata.csv")
- jul2021 <- read_csv("/Users/…/202107-divvy-tripdata.csv")
- aug2021 <- read_csv("/Users/…/202108-divvy-tripdata.csv")
- sep2021 <- read_csv("/Users/…/202109-divvy-tripdata.csv")
- oct2021 <- read_csv("/Users/…/202110-divvy-tripdata.csv")
- nov2021 <- read_csv("/Users/…/202111-divvy-tripdata.csv")
- dec2021 <- read_csv("/Users/…/202112-divvy-tripdata.csv")

Inspect and cleaning data

Compare_df_cols_same: Do the the data.frames have the same columns & types?

I learnt a lot from people who shared their codes online. This step and the row bind step I especially am grateful towards [Alessandro Ferrarese] (https://www.alessandroferrarese.com/ (https://www.alessandroferrarese.com/))

```
compare_df_cols_same(jan2021,feb2021,mar2021,apr2021,may2021,jun2021,jul2021,aug2021,se
p2021,oct2021,nov2021,dec2021,
                     bind_method = c("bind_rows", "rbind"), verbose = TRUE)
```

```
## [1] TRUE
```

The result was true, means the columns number was the same and the types were also similar, that I could proceed to use row bind commands to bind my different data sets together.

result equals TRUE, proceed to binding

```
tripdata_2021 <- rbind(jan2021,feb2021,mar2021,apr2021,may2021,jun2021,jul2021,aug2021,
sep2021,oct2021,nov2021,dec2021)
```

```
str(tripdata_2021)
```

```
## spc_tbl_ [5,595,063 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5595063] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C9
4683FE3F27" "4FA453A75AE377DB" ...
##  $ rideable_type     : chr [1:5595063] "electric_bike" "electric_bike" "electric_bik
e" "electric_bike" ...
##  $ started_at        : POSIXct[1:5595063], format: "2021-01-23 16:14:19" "2021-01-27
18:43:08" ...
##  $ ended_at          : POSIXct[1:5595063], format: "2021-01-23 16:24:44" "2021-01-27
18:47:12" ...
##  $ start_station_name: chr [1:5595063] "California Ave & Cortez St" "California Ave
& Cortez St" "California Ave & Cortez St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:5595063] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:5595063] NA NA NA NA ...
##  $ end_station_id    : chr [1:5595063] NA NA NA NA ...
##  $ start_lat         : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:5595063] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
glimpse(tripdata_2021)
```

```
## Rows: 5,595,063
## Columns: 13
## $ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683…
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", …
## $ started_at         <dttm> 2021-01-23 16:14:19, 2021-01-27 18:43:08, 2021-01-…
## $ ended_at           <dttm> 2021-01-23 16:24:44, 2021-01-27 18:47:12, 2021-01-…
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor…
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660…
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augu…
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258",…
## $ start_lat          <dbl> 41.90034, 41.90033, 41.90031, 41.90040, 41.90033, 4…
## $ start_lng          <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -87.696…
## $ end_lat            <dbl> 41.89000, 41.90000, 41.90000, 41.92000, 41.90000, 4…
## $ end_lng            <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -87.700…
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "…
```

Remove unnecessary columns, the geographical specifications in columns from 9 to 12

```
trimmed_tripdata_2021 <- tripdata_2021[-c(9:12)]
glimpse(trimmed_tripdata_2021)
```

```
## Rows: 5,595,063
## Columns: 9
## $ ride_id            <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683…
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", …
## $ started_at         <dttm> 2021-01-23 16:14:19, 2021-01-27 18:43:08, 2021-01-…
## $ ended_at           <dttm> 2021-01-23 16:24:44, 2021-01-27 18:47:12, 2021-01-…
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor…
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660…
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augu…
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258",…
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "…
```

Remove rows with NA and NULL values

First I went on to check for the total number of rows with NA and NULL values using “is.na” and “is.null”

```
sum(is.na(trimmed_tripdata_2021))
```

```
## [1] 2859955
```

```
sum(is.null(trimmed_tripdata_2021))
```

```
## [1] 0
```

The result showed that there were 2859955 rows with NA values and 0 with NULL values. Proceed to clean the NA rows with “drop_na”

```
clean_tripdata <- trimmed_tripdata_2021 %>% drop_na()
glimpse(clean_tripdata)
```

```
## Rows: 4,588,302
## Columns: 9
## $ ride_id            <chr> "B9F73448DFBE0D45", "457C7F4B5D3DA135", "57C750326F…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "…
## $ started_at         <dttm> 2021-01-24 19:15:38, 2021-01-23 12:57:38, 2021-01-…
## $ ended_at           <dttm> 2021-01-24 19:22:51, 2021-01-23 13:02:10, 2021-01-…
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor…
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660…
## $ end_station_name   <chr> "Wood St & Augusta Blvd", "California Ave & North A…
## $ end_station_id     <chr> "657", "13258", "657", "657", "657", "KA1504000135"…
## $ member_casual      <chr> "member", "member", "casual", "casual", "casual", "…
```

```
sum(is.na(clean_tripdata))
```

```
## [1] 0
```

After cleaning, counted again but this time there was no more NA rows.

Process and modify data

Rename column for better understanding later when doing further analysis or visualization

```
clean_tripdata <- clean_tripdata %>% rename(customer_type=member_casual)
glimpse(clean_tripdata)
```

```
## Rows: 4,588,302
## Columns: 9
## $ ride_id            <chr> "B9F73448DFBE0D45", "457C7F4B5D3DA135", "57C750326F…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "…
## $ started_at         <dttm> 2021-01-24 19:15:38, 2021-01-23 12:57:38, 2021-01-…
## $ ended_at           <dttm> 2021-01-24 19:22:51, 2021-01-23 13:02:10, 2021-01-…
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor…
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660…
## $ end_station_name   <chr> "Wood St & Augusta Blvd", "California Ave & North A…
## $ end_station_id     <chr> "657", "13258", "657", "657", "657", "KA1504000135"…
## $ customer_type      <chr> "member", "member", "casual", "casual", "casual", "…
```

Calculate length of rides in minutes and add to the existing data.frame as a new column

The datasets provided us with start timestamp and end timestamp so we could try to find the duration of the ride. And later, we can also extract time, date, and month from the start timestamp to compare between two groups, members and casuals.

```
ride_length <- difftime(clean_tripdata$ended_at,clean_tripdata$started_at,units = "min
s")
clean_tripdata$ride_length_mins = ride_length
glimpse(clean_tripdata)
```

```
## Rows: 4,588,302
## Columns: 10
## $ ride_id            <chr> "B9F73448DFBE0D45", "457C7F4B5D3DA135", "57C750326F…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "…
## $ started_at         <dttm> 2021-01-24 19:15:38, 2021-01-23 12:57:38, 2021-01-…
## $ ended_at           <dttm> 2021-01-24 19:22:51, 2021-01-23 13:02:10, 2021-01-…
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor…
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660…
## $ end_station_name   <chr> "Wood St & Augusta Blvd", "California Ave & North A…
## $ end_station_id     <chr> "657", "13258", "657", "657", "657", "KA1504000135"…
## $ customer_type      <chr> "member", "member", "casual", "casual", "casual", "…
## $ ride_length_mins   <drtn> 7.216667 mins, 4.533333 mins, 9.783333 mins, 8.950…
```

Check for rides that were quality check and for negative values in ride_length_mins

I wanted to check for rides that were: (1) conducted by the company as quality tests, with “TEST” as start station name (2) ride lengths were negative owing to calculation errors

As those data could bias the analysis result

```
nrow(subset(clean_tripdata, start_station_name == "TEST"))
```

```
## [1] 0
```

```
nrow(subset(clean_tripdata, ride_length_mins < 0))
```

```
## [1] 116
```

There was no quality ride but 116 rides with negative results

Remove negative values in ride_length_mins

```
clean_tripdata <- filter(clean_tripdata, ride_length_mins >=1)
nrow(subset(clean_tripdata, ride_length_mins < 0))
```

```
## [1] 0
```

No we have no more negative ride lengths

Extract specifics from the initiation time-stamp for subsequent analysis: the start hour, the start day of the week, and the start month

Using functions from the “lubridate” package

```
start_hour <- hour(clean_tripdata$started_at)
clean_tripdata$ride_hour = start_hour

start_wday <- wday(clean_tripdata$started_at, TRUE)
clean_tripdata$ride_wday = start_wday

start_month <- month(clean_tripdata$started_at, label=TRUE, abbr=FALSE)
clean_tripdata$ride_month = start_month

glimpse(clean_tripdata)
```

```
## Rows: 4,528,933
## Columns: 13
## $ ride_id            <chr> "B9F73448DFBE0D45", "457C7F4B5D3DA135", "57C750326F…
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "…
## $ started_at         <dttm> 2021-01-24 19:15:38, 2021-01-23 12:57:38, 2021-01-…
## $ ended_at           <dttm> 2021-01-24 19:22:51, 2021-01-23 13:02:10, 2021-01-…
## $ start_station_name <chr> "California Ave & Cortez St", "California Ave & Cor…
## $ start_station_id   <chr> "17660", "17660", "17660", "17660", "17660", "17660…
## $ end_station_name   <chr> "Wood St & Augusta Blvd", "California Ave & North A…
## $ end_station_id     <chr> "657", "13258", "657", "657", "657", "KA1504000135"…
## $ customer_type      <chr> "member", "member", "casual", "casual", "casual", "…
## $ ride_length_mins   <drtn> 7.216667 mins, 4.533333 mins, 9.783333 mins, 8.950…
## $ ride_hour          <int> 19, 12, 15, 15, 15, 15, 10, 11, 7, 8, 8, 13, 9, 11,…
## $ ride_wday          <ord> Sun, Sat, Sat, Sat, Sun, Fri, Tue, Sat, Wed, Fri, S…
## $ ride_month         <ord> January, January, January, January, January, Januar…
```

Now we can export the cleaned and processed data for further analysis or visualization.

#Export the data.set for further analysis write.csv(clean_tripdata, “finalfinal_trip_data_2021.csv”)